12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
10 - 12 September 2008, Victoria, British Columbia Canada

AIAA 2008-5964

A Compact Proper Orthogonal Decomposition Basis for Optimization-Oriented Reduced-Order Models

Kevin Carlberg* and Charbel Farhat†
Stanford University, Stanford, CA, 94305-3035, USA

Reduced basis methods are powerful tools that can significantly speed up computationally expensive analyses in a variety of "many-query" and real-time applications, including design optimization. Unfortunately, these techniques produce reduced-order models (ROMs) that are costly to construct and are not always robust in the parameter space. Furthermore, efforts to apply these methods to optimization have almost exclusively relied on offline sampling procedures, which can exacerbate these problems and even prevent computational cost savings from being realized. Although adaptive sampling techniques have been recently developed to avoid these drawbacks, their application has been limited to the optimization of dynamical systems. We therefore present a Compact POD (CPOD) basis that treats the (possibly high-order) state vector sensitivities as system "snapshots" and employs a snapshot-weighting scheme to compute a reduced basis. Since sensitivities directly represent the effect of parameter changes on the system's physics and can often be efficiently computed, the proposed method produces an inherently robust, inexpensive ROM that is well-suited for optimization. We also show that it can be used to extend adaptive sampling procedures to static systems. To test the validity of the CPOD basis, we compare it with the typical POD approach on the analysis of a plane elasticity problem parameterized by shape and material property variables. We find that the proposed methodology leads to solutions of similar (often better) accuracy, at a lower cost, and using many fewer reference configurations than the typical procedure.

Nomenclature

a	Symmetric, bilinear, coercive, parameterized operator	$\mathcal{E}$	Fraction of retained energy in POD basis
		f	Linear operator
c	Equality constraint	F	Finite element force vector
γ	Snapshot weight	h	Domain height
$\gamma^{\mathcal{T}}$	Taylor expansion weights	$\mathcal{H}$	Hermite subspace
$\gamma^{\mathcal{D}}$	Configuration distance weight	J	Objective function
Γ	Diagonal matrix of snapshot weights	$\mathcal{J}$	Reduced Jacobian
d	Inequality constraint	K	Finite element stiffness matrix
δ	Relative projection error	L	Domain length
$\delta_{\min}, \bar{\delta}$	Minimum and average δ	m	Order of Compact POD basis
Δy	Vertial displacement of control node	$\boldsymbol{\mu}$	Vector of parameters
$\mathcal{D}$	Parameter domain	$\boldsymbol{\mu}^i$	$\boldsymbol{\mu}$ at i^{th} reference configuration
$\mathcal{D}_r, \tilde{\mathcal{D}}_r$	Sets of reference configurations	$\bar{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$ at target configuration
e	Error of approximate solution	μ_i	i^{th} entry of $\boldsymbol{\mu}$
$e_{\parallel}$, $e_{\perp}$	Component of e in, orthogonal to $\mathcal{P}(n_\phi)$	ν	Poisson's ratio
e_r	Relative error in energy norm	n_{ec}, n_{ic}	Number of equality, inequality constraints
E	Modulus of elasiticity	n_μ	Number of parameters

*Graduate Student, Department of Aeronautics & Astronautics, Building 500, 488 Escondido Mall, Stanford University, Stanford, CA 94305-3035, AIAA Member.
†Vivian Church Hoff Professor of Aircraft Structures, Department of Aeronautics & Astronautics, William F. Durand Building, Room 257, Stanford University, Stanford, CA 94305-4035, AIAA Fellow.

n_r	Number of reference configurations
n_s	Number of snapshots
$n_C(m)$	Number of snapshots for Compact POD of order m
n_ϕ	Number of retained POD basis vectors
$n_{\phi,\mathcal{E}}$	n_ϕ as determined by energy criterion
N	Dimension of full-order model
ξ	Finite element basis function
p	Physical parameter
$\mathcal{P}(n_\phi)$	POD subspace of dimension n_ϕ
$\mathcal{R}$	Operator defining static equilibrium
ρ	Material density
$\bar{\sigma}$	Singular value of $\bar{W}$
$\bar{\Sigma}$	Diagonal matrix of singular values $\bar{\sigma}$
$\mathcal{S}$	Space of trial functions
t	Plate thickness
$\mathcal{T}$	Subspace for POD optimization problem
u	State vector
$\hat{u}$	POD modal coordinates
u, v	Trial, test functions
υ	Fraction of retained energy in POD basis
$\mathcal{V}$	Subspace used in determining $n_{\phi,\mathcal{E}}$
ϕ	POD basis vector
Φ	POD basis of implied dimension
$\Phi(n_\phi)$	POD basis of dimension n_ϕ
$\bar{\Phi}$, $\bar{\Psi}$	Left and right singular vectors of $\bar{W}$
w	Snapshot
W	Weighted snapshot matrix
$\bar{W}$	Transformed snapshot matrix
$\tilde{W}(\boldsymbol{\mu})$	Matrix of unweighted snapshots computed for parameters $\boldsymbol{\mu}$
ω	Configuration distance weighting exponent
Ω	Problem domain
$\partial\Omega_g$	Portion of boundary with Dirichlet conditions
Subscripts	
C^m	Compact POD of order m
POD, s	Typical POD basis, same n_s as Compact POD
POD, r	Typical POD basis, same n_r as Compact POD
i, j, k, l	Integer-valued indices
Superscripts	
h	Finite element
Other	
$\tilde{\cdot}$	Reduced-order model
$(\cdot,\cdot)_{E,\boldsymbol{\mu}}$	Energy inner product, continuous arguments
$(\cdot,\cdot)_{K(\boldsymbol{\mu})}$	Energy inner product, discrete arguments
$\|\cdot\|_{E,\boldsymbol{\mu}}$	Energy norm, continuous arguments
$\|\cdot\|_{K(\boldsymbol{\mu})}$	Energy norm, discrete arguments

I. Introduction

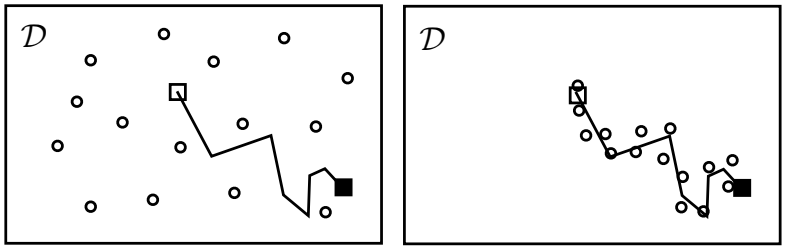
REDUCED basis approximation leads to a class of model-reduction methods that can dramatically decrease the computational cost of solving large, parameterized systems in one of two situations: the "many-query" context and the real-time context. The many-query setting arises when the system is solved many times for various configurations, as occurs in design optimization,[1,2] optimal control,[3–6] multi-scale simulation,[7,8] and uncertainty analysis.[9,10] The real-time framework is often desired for routine analysis,[11–13] control,[14–18] and nondestructive evaluation/parameter estimation,[19–21] for example. In these scenarios, reduced basis methods employ previously-computed parameter-induced system realizations, or "snapshots,"[22] to generate a basis of relatively small dimension. The governing equations are then projected onto the corresponding subspace, resulting in a much smaller system referred to as a reduced-order model (ROM). Proper Orthogonal Decomposition (POD), which is also known in the literature as Karhunen-Loève expansion and principal component analysis, is a specific type of reduced basis technique closely connected with the singular value decomposition (SVD) of a matrix. This method is characterized by its ability to optimally truncate the basis such that it represents only the most energetic modes contained in the snapshots. POD has gained widespread attention recently due to this optimality property.

In this work, we consider the design optimization of static systems, which falls in the many-query category. To apply reduced basis approximation to this problem, the approach currently taken in the literature is to adapt an *offline-online* strategy.[1,2] In this approach, the parameter space is first sampled for snapshots and the reduced basis is computed in the *offline* phase, and design optimization using the resulting ROM is carried out in the *online* phase.[a] While this strategy is necessary for real-time online computations, it carries several drawbacks in the many-query context which can prevent desired accuracy and savings in computational cost from being achieved. Firstly, it suffers from a 'break-even' point,[2,8,18,24] which is characterized by the number of online system and gradient evaluations that are required before computational cost savings for the overall offline-online procedure are obtained. Namely, if more system solutions are computed in the offline sampling phase than system and gradient solutions are required online, it is computationally less

[a] This is also the most popular approach for the design optimization of dynamical systems, where the sampling phase can have the broader interpretation of applying a generalized excitation to the system,[14,18] or computing multiple snapshots at a single configuration.[4,23]

expensive (and more accurate) to simply optimize using the full-order model (FOM) and bypass reduced basis approximation entirely.

Secondly, since the offline phase occurs before the online optimization takes place, the sampling procedure (common approaches include uniform, random, stratified,[25] centroidal Voronoi tesselation,[26] and greedy[24,27,28] sampling) cannot take the (unknown) optimization trajectory into account. As a result, snapshots will be computed at points in the parameter space far from the optimizer's path—a waste of computational resources. This is illustrated in figure 1(a). If we knew the optimization trajectory *a priori*, we would cluster the samples around it as shown in figure 1(b) to improve accuracy in our approximation. To



(a) Offline Sampling (b) "Ideal" Sampling

Figure 1. Sampling for the optimization of static systems. $\mathcal{D}$ parameter space, — optimization trajectory, □ initial parameters, ■ optimal parameters, ∘ sample parameters.

circumvent this problem, several adaptive sampling procedures have been developed in the optimal control setting, including Trust Region POD (TRPOD),[5] POD for Optimality Systems (OS-POD),[6] and others.[3,4] These methods update or replace the reduced basis as the optimization algorithm progresses, but assume that a snapshot *ensemble* can be computed at a single configuration for a given problem. Although this assumption holds for dynamical systems where snapshots correspond to the state vector at multiple times or frequencies, it does *not* hold for static systems, where only one snapshot (the steady-state solution) can be obtained for fixed boundary conditions. Thus, these adaptive procedures cannot be applied to static systems using existing techniques.

Furthermore, reduced basis approximations are, in general, not robust. In the case of a POD basis constructed at a single point in the parameter space, the basis is optimal only for this configuration and contains no information about the physics induced by parameter changes. In the context of dynamical systems, POD subspace interpolation[12,13] and the basis updating procedures mentioned above have been successfully applied to overcome this shortcoming, but their application to static problems is again complicated by the limitation of one snapshot per configuration. The aforementioned parameter space sampling procedures address this problem to some degree for static systems, but the resulting basis contains no information *per se* about changes in the physics caused by arbitrary parameter variations.

Lastly, the offline phase is very computationally expensive, because each sample requires a reconfiguration of the system and the computation of a steady-state solution. The former may require a mesh deformation, which can be particularly expensive in fluids applications, for example. The latter will also be costly, since a completely new set of governing equations must be solved for each snapshot, thereby precluding the use of efficient multiple-RHS techniques.

In this work, we present an enhanced POD basis that addresses the above problems and is particularly well-suited for the design optimization of static systems. In essence, we extend the notion of a Hermite basis,[16] which uses the solutions and gradients (sensitivities) with respect to the parameters as a basis, to higher order derivatives and develop a snapshot-weighting scheme that allows for an optimal truncation of the basis. The result is an inexpensive, robust, design-oriented basis that can be easily implemented in existing gradient-based optimization software. We refer to this method as *Compact POD* (CPOD). We also propose an extension of adaptive sampling to the design optimization of static systems, where the CPOD basis is a critical component.

II. Reduced Order Modeling via Proper Orthogonal Decomposition

In this section, we outline some properties of Proper Orthogonal Decomposition and discuss its application to model reduction of discretized, static systems.

II.A. POD and the optimal truncation property

POD computes a basis that optimally represents a given set of snapshots. The snapshots typically correspond to solutions (state vector realizations) of the system at various times, frequencies, or configurations.

Assume we have computed a collection of n_s snapshots $w_i \in \mathbb{R}^N$, $1 \leq i \leq n_s$, where N is the (large) number of degrees of freedom in the full-order model (FOM). We would like to compute a truncated basis $\Phi(n_\phi) = \left[\phi_1, \ldots, \phi_{n_\phi}\right] \in \mathbb{R}^{N \times n_\phi}$ with $n_\phi \leq n_s$ that optimally represents these snapshots. The vectors $\phi_i \in \mathbb{R}^N$, $1 \leq i \leq n_\phi$ can be determined by the following sequence of constrained, weighted least-squares problems:

$$\begin{aligned} \phi_i &= \arg\min_{\psi \in \mathcal{T}_i} \sum_{j=1}^{n_s} \gamma_j^2 \|w_j - (w_j, \psi)_\Theta \, \psi\|_\Theta^2 \\ \mathcal{T}_i &\equiv \{\psi \in \mathbb{R}^N \mid \|\psi\|_\Theta = 1, \ (\psi, \phi_k)_\Theta = 0 \ \ \forall k < i\}. \end{aligned} \tag{1}$$

Here, $\gamma_j \in \mathbb{R}$, $1 \leq j \leq n_s$ are snapshot weights, and $(\cdot, \cdot)_\Theta$ is a weighted inner product defined on the Euclidean space $\mathbb{R}^N$ as $(u, v)_\Theta \equiv u^T \Theta v$. The matrix $\Theta \in \mathbb{R}^{N \times N}$ is symmetric and positive definite with a Cholesky decomposition $\Theta = (\Theta^{1/2})^T \Theta^{1/2}$. The norm induced by this inner product is defined as $\|u\|_\Theta \equiv \sqrt{u^T \Theta u}$. Note that $(\phi_i, \phi_j)_\Theta = \delta_{ij}$.

Fortunately, this sequence of $\Theta-$orthogonal vectors can be easily computed using the thin SVD.[29] To do this, we first choose weights γ_j, $1 \leq j \leq n_s$ and assemble the weighted snapshot matrix $W = [\gamma_1 w_1, \ldots, \gamma_{n_s} w_{n_s}] \in \mathbb{R}^{N \times n_s}$. Then, we use this matrix in algorithm 1.

Algorithm 1 POD basis computation given W.

1: $\bar{W} \leftarrow \Theta^{1/2} W$
2: $\bar{W} = \bar{\Phi} \bar{\Sigma} \bar{\Psi}^T$ {Compute Thin SVD}
3: $\Phi(n_s) \leftarrow \Theta^{-1/2} \bar{\Phi}$, where $\Phi(n_s) = [\phi_1, \ldots, \phi_{n_s}]$
4: Choose size of truncated basis n_ϕ.
5: $\Phi(n_\phi) = \left[\phi_1, \ldots, \phi_{n_\phi}\right]$ {Truncate}

The resulting basis $\Phi(n_\phi)$ is referred to as the POD basis, and it is optimal in the sense of Eq. (1). In other words, the POD subspace $\mathcal{P}(n_\phi) \equiv \text{span}\{\phi_i, 1 \leq i \leq n_\phi\}$ minimizes the weighted sum of squares of the normed projection errors of the snapshots onto a subspace of dimension n_ϕ. In fact, this error can be computed directly from step 2 of algorithm 1 as

$$\sum_{j=1}^{n_s} \gamma_j^2 \|w_j - \sum_{i=1}^{n_\phi} (w_j, \phi_i)_\Theta \, \phi_i\|_\Theta^2 = \sum_{k=n_\phi+1}^{n_s} \bar{\sigma}_k^2. \tag{2}$$

Here, $\bar{\sigma}_k$ is the k^{th} diagonal entry of $\bar{\Sigma}$ (the k^{th} largest singular value of $\bar{W}$). This error can be considered the "energy" of the snapshots omitted by the POD basis. We therefore define the relative energy retained by the basis as

$$\mathcal{E}(\bar{\Sigma}, n_\phi) \equiv \frac{\sum_{i=1}^{n_\phi} \bar{\sigma}_i^2}{\sum_{j=1}^{n_s} \bar{\sigma}_j^2}. \tag{3}$$

Often, we choose n_ϕ in step 4 such that $\mathcal{E}\left(\bar{\Sigma}, n_\phi\right)$ is greater than some threshold. Setting $\upsilon \in [0,1]$ to the fraction of total energy we would like the reduced basis to retain, we can choose $n_\phi = n_{\phi,\mathcal{E}}\left(\bar{\Sigma}, \upsilon\right)$, where

$$\begin{aligned} n_{\phi,\mathcal{E}}\left(\bar{\Sigma}, \upsilon\right) &\equiv \min_{n \in \mathcal{V}\left(\bar{\Sigma}, \upsilon\right)} n \\ \mathcal{V}\left(\bar{\Sigma}, \upsilon\right) &\equiv \{n \in \{1, 2, \ldots, n_s\} \mid \mathcal{E}\left(\bar{\Sigma}, n\right) \geq \upsilon\}. \end{aligned} \tag{4}$$

Thus, $n_{\phi,\mathcal{E}}\left(\bar{\Sigma}, \upsilon\right)$ is the smallest dimension for which the relative energy retained by the POD basis exceeds some fraction υ. Typically, $\upsilon \in [0.9, 1)$.

II.B. Galerkin projection

Since we are focusing on static systems, consider the system of equations

$$\mathcal{R}\left(u\left(\boldsymbol{\mu}\right); \boldsymbol{\mu}\right) = 0 \tag{5}$$

corresponding to the full-order model of a (general) parameterized, discretized, static system. Here, the state vector $u\left(\boldsymbol{\mu}\right) \in \mathbb{R}^N$ corresponds to the full-order solution, whose dependence on the system parameters $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_{n_\mu}\right) \in \mathcal{D} \subset \mathbb{R}^{n_\mu}$ is enforced by the operator $\mathcal{R} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$ through Eq. (5). The closed, bounded parameter domain is denoted by $\mathcal{D}$. To reduce the system to dimension $n_\phi \ll N$, we project Eq. (5) onto the POD subspace $\mathcal{P}\left(n_\phi\right)$ via Galerkin projection to obtain a smaller system of equations associated with the reduced-order model:

$$\Phi^T \mathcal{R}\left(\Phi\hat{u}; \boldsymbol{\mu}\right) = 0. \tag{6}$$

If $\mathcal{R}$ represents a nonlinear operator in u, we can solve Eq. (6) by Newton's method (for example) given a configuration (represented by $\boldsymbol{\mu}$) as

$$\begin{aligned} \left[\Phi^T \frac{\partial \mathcal{R}}{\partial u}\left(\Phi\hat{u}^{(k)}; \boldsymbol{\mu}\right)\Phi\right]\delta\hat{u}^{(k)} &= -\Phi^T \mathcal{R}\left(\Phi\hat{u}^{(k)}; \boldsymbol{\mu}\right) \\ \hat{u}^{(k+1)} &= \hat{u}^{(k)} + \delta\hat{u}^{(k)}, \end{aligned} \tag{7}$$

where $\hat{u}^{(k)} \in \mathbb{R}^{n_\phi}$ are the reduced coordinates at iteration k, and the bracketed quantity $\mathcal{J}\left(\hat{u}^{(k)}, \Phi, \boldsymbol{\mu}\right) \equiv \left[\Phi^T \frac{\partial \mathcal{R}}{\partial u}\left(\Phi\hat{u}^{(k)}; \boldsymbol{\mu}\right)\Phi\right]$ with $\mathcal{J} : \mathbb{R}^{n_\phi} \times \mathbb{R}^{N \times n_\phi} \times \mathcal{D} \to \mathbb{R}^{n_\phi \times n_\phi}$ is the reduced Jacobian.

If $\mathcal{R}$ represents a linear operator in u, we can solve Eq. (6) in a single step:

$$\left[\Phi^T \frac{\partial \mathcal{R}}{\partial u}\left(\boldsymbol{\mu}\right)\Phi\right]\hat{u} = \Phi^T \mathcal{R}\left(0; \boldsymbol{\mu}\right). \tag{8}$$

In either case, an approximate solution can be recovered (with $\hat{u} = \hat{u}^{(K+1)}$ from the final iteration in the nonlinear case) by

$$\tilde{u}\left(\Phi, \boldsymbol{\mu}\right) = \Phi\hat{u}. \tag{9}$$

By construction, $\tilde{u}\left(\Phi, \boldsymbol{\mu}\right) \in \mathcal{P}\left(n_\phi\right)$, which clearly limits the accuracy of our approximate solution. We can define the error of this approximation as

$$e\left(\Phi, \boldsymbol{\mu}\right) = u\left(\boldsymbol{\mu}\right) - \tilde{u}\left(\Phi, \boldsymbol{\mu}\right). \tag{10}$$

This can be decomposed into orthogonal components:[30]

$$e\left(\Phi, \boldsymbol{\mu}\right) = e_{\parallel}\left(\Phi, \boldsymbol{\mu}\right) + e_{\perp}\left(\Phi, \boldsymbol{\mu}\right) \tag{11}$$

$$e_{\parallel}\left(\Phi, \boldsymbol{\mu}\right) = \sum_{i=1}^{n_\phi} \left(\phi_i, u\left(\boldsymbol{\mu}\right)\right)_\Theta \phi_i - \tilde{u}\left(\Phi, \boldsymbol{\mu}\right) \tag{12}$$

$$e_{\perp}\left(\Phi, \boldsymbol{\mu}\right) = u\left(\boldsymbol{\mu}\right) - \sum_{i=1}^{n_\phi} \left(\phi_i, u\left(\boldsymbol{\mu}\right)\right)_\Theta \phi_i. \tag{13}$$

Note that $e_{\|}(\Phi, \boldsymbol{\mu}) \in \mathcal{P}(n_\phi)$ and $e_{\perp}(\Phi, \boldsymbol{\mu}) \perp \mathcal{P}(n_\phi)$ by construction. We therefore seek a basis Φ that minimizes the projection error of the full-order solution onto its subspace $\left(\text{small } e_{\perp}(\Phi, \boldsymbol{\mu})\right)$ and a solution method that minimizes the error between the approximate solution and the projection of the full-order solution onto this subspace $\left(\text{small } e_{\|}(\Phi, \boldsymbol{\mu})\right)$. By comparing Eqs. (1) and (13), we see that the POD basis meets the first of these objectives, since it minimizes a weighted sum of squares of $\|e_{\perp}\left(\Phi, \boldsymbol{\mu}^j\right)\|_\Theta$ when the snapshots correspond to the state vector at various configurations $\left(w_j = u\left(\boldsymbol{\mu}^j\right), 1 \leq j \leq n_s\right)$.

III. Compact POD

We now present the development of the Compact POD basis, which we show is well-suited for the design optimization of static systems.

III.A. Hermite subspace

The Hermite subspace was introduced in Ref. [16] as a generalization of the Taylor subspace[31,32] in the context of optimally controlling a fluid system in steady-state using a single parameter (the Reynolds number). This method uses the subspace spanned by the solution u and its first-order derivative (sensitivity) with respect to a single parameter μ at n_r distinct parameter values to generate a ROM via Galerkin projection. Specifically, the Hermite subspace is defined as

$$\mathcal{H} \equiv \text{span}\{u\left(\mu^i\right), \frac{du}{d\mu}\left(\mu^i\right), 1 \leq i \leq n_r\}. \tag{14}$$

In spite of the promising results[b] presented in Ref. [16], this method has received little attention in the literature. This is likely due to the pervasiveness of dynamical systems in controls applications and a limitation of the method—it does not allow for optimal truncation in a straightforward manner.

To illustrate the second point, define the snapshot matrix in the obvious way as

$$W = \left[u\left(\mu^1\right), \frac{du}{d\mu}\left(\mu^1\right), \ldots, u\left(\mu^{n_r}\right), \frac{du}{d\mu}\left(\mu^{n_r}\right)\right], \tag{15}$$

where $n_s = 2n_r$. If we use Eq. (15) in algorithm 1, the resulting POD basis will be essentially meaningless. This is because the magnitude of the snapshots corresponding to the sensitivities can be arbitrarily changed by scaling the parameter $\mu \leftarrow \alpha\mu$ by any $\alpha \in \mathbb{R}$. Since the truncated POD basis is very sensitive to the magnitude of the snapshots $\left(\text{via the inner product in Eq. (1)}\right)$, we obtain the following results for $n_\phi = n_r$:

$$\begin{aligned} \lim_{\alpha \to 0} \mathcal{P}(n_r) &= \text{span}\{u\left(\mu^i\right), 1 \leq i \leq n_r\}, \\ \lim_{\alpha \to \infty} \mathcal{P}(n_r) &= \text{span}\{\frac{du}{d\mu}\left(\mu^i\right), 1 \leq i \leq n_r\}. \end{aligned} \tag{16}$$

Thus, the Hermite snapshot matrix in Eq. (15) produces a POD subspace that is entirely dependent on an arbitrary scaling factor. So, in its existing form, we cannot employ POD-based truncation with the Hermite subspace and we must use $\mathcal{H}$ of (full) dimension n_s for model reduction. This is a significant drawback, since POD typically enables the subspace spanned by all snapshots to be reduced to dimension $n_\phi \ll n_s$ in an optimal sense.

We therefore proceed by proposing a method that improves upon the Hermite subspace. First, we generalize the Hermite subspace approach to exploit higher-order sensitivities. Next, we introduce a snapshot-weighting scheme that enables the computation of a POD basis that can be meaningfully truncated. We refer to the resulting basis as the *Compact POD* (CPOD) basis. We also outline opportunities for efficiently computing the CPOD snapshots and use CPOD to extend adaptive sampling procedures to the optimization of static systems.

[b]The authors found the Hermite subspace to produce more accurate results using fewer reference configurations than the Lagrange subspace, which uses state vector solutions as basis vectors.

III.B. Taylor expansion weights

To motivate the first component of our weighting scheme, consider an arbitrary number of parameters n_μ, a single reference configuration ($n_r = 1$) defined by $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_{n_\mu}\right) \in \mathcal{D}$ where the solution and sensitivities to order m have been computed, and a target configuration defined by $\bar{\boldsymbol{\mu}} = \left(\bar{\mu}_1, \ldots, \bar{\mu}_{n_\mu}\right) \in \mathcal{D}$. This target can be given, for example, by the next iteration in an optimization algorithm, and it should be near the point in $\mathcal{D}$ where we will compute an approximate solution. Treating the solution and sensitivities as snapshots, we assemble the weighted snapshot matrix W as

$$\tilde{W}(\boldsymbol{\mu}, m) \equiv \left[\tilde{w}_1(\boldsymbol{\mu}), \ldots, \tilde{w}_{n_C(m)}(\boldsymbol{\mu})\right] \tag{17}$$

$$\tilde{w}_j\left(\boldsymbol{\mu}\right) \equiv \begin{cases} u(\boldsymbol{\mu}), & j = 1 \\ \frac{\partial u}{\partial \mu_{j-1}}(\boldsymbol{\mu}), & 2 \leq j \leq n_C(1) \\ \frac{\partial^2 u}{\partial \mu_k \partial \mu_l}(\boldsymbol{\mu}), & n_C(1)+1 \leq j \leq n_C(2),\ 1 \leq k \leq l \leq n_\mu \\ \vdots & \vdots \end{cases} \tag{18}$$

$$\Gamma(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}, m) = \operatorname{diag}\left(\gamma_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}\right)\right) \tag{19}$$

$$W = \tilde{W}(\boldsymbol{\mu}, m)\Gamma(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}, m), \tag{20}$$

where $n_C(m)$ is the number of (unique) snapshots collected at one configuration when sensitivities to order m are computed:

$$n_C\left(m\right) = \begin{cases} 1, & m = 0 \\ n_\mu + 1, & m = 1 \\ \frac{1}{2}n_\mu^2 + \frac{3}{2}n_\mu + 1, & m = 2 \\ \vdots & \vdots \end{cases} \tag{21}$$

Before computing the POD basis via algorithm 1, we must determine appropriate weights γ_j, $1 \leq j \leq n_C(m)$. Note that weights of unity can lead to a meaningless truncated basis (see Section III.A).

Assuming that the state vector $u : \mathcal{D} \to \mathbb{R}^N$ is $m+1$ times continuously differentiable, we can use the Taylor expansion of the state vector about $\boldsymbol{\mu}$ to write:

$$\begin{aligned} u\left(\bar{\boldsymbol{\mu}}\right) =& u\left(\boldsymbol{\mu}\right) + \sum_{i=1}^{n_\mu}\left(\bar{\mu}_i - \mu_i\right)\frac{\partial u}{\partial \mu_i}\left(\boldsymbol{\mu}\right) + \sum_{i=1}^{n_\mu}\sum_{j=1}^{n_\mu}\frac{1}{2}\left(\bar{\mu}_i - \mu_i\right)\left(\bar{\mu}_j - \mu_j\right)\frac{\partial^2 u}{\partial \mu_i \partial \mu_j}\left(\boldsymbol{\mu}\right) \\ &+ \cdots + \sum_{i=1}^{n_\mu}\cdots\sum_{j=1}^{n_\mu}\frac{1}{m!}\left(\bar{\mu}_i - \mu_i\right)\cdots\left(\bar{\mu}_j - \mu_j\right)\frac{\partial^m u}{\partial \mu_i \cdots \partial \mu_j}\left(\boldsymbol{\mu}\right) + \mathcal{O}\left(\left\|\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}\right\|_2^{m+1}\right). \end{aligned} \tag{22}$$

The snapshot weights required to recover $u\left(\bar{\boldsymbol{\mu}}\right)$ to order $m+1$ are apparent from Eq. (22):

$$\gamma_j^{\mathcal{T}}\left(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}\right) \equiv \begin{cases} 1, & j = 1 \\ \left(\bar{\mu}_{j-1} - \mu_{j-1}\right), & 2 \leq j \leq n_C(1) \\ \left(1 - \frac{1}{2}\delta_{kl}\right)\left(\bar{\mu}_k - \mu_k\right)\left(\bar{\mu}_l - \mu_l\right), & n_C(1)+1 \leq j \leq n_C(2),\ 1 \leq k \leq l \leq n_\mu \\ \vdots & \vdots \end{cases} \tag{23}$$

Using these *Taylor expansion weights* $\gamma_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}\right) = \gamma_j^{\mathcal{T}}\left(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}\right)$ in Eqs. (17)–(20) and algorithm 1, we can compute a POD basis. This basis is optimally truncated in the sense of the $(m+1)$-order Taylor approximation of the solution at $\bar{\boldsymbol{\mu}}$ given the data at $\boldsymbol{\mu}$.

III.C. Configuration distance weights

At first glance, the extension of the results of Section III.B to multiple reference configurations ($n_r > 1$) seems straightforward, since we can simply combine the weighted snapshot matrices of Eq. (20) from each

reference configuration:

$$W = \left[\tilde{W}(\boldsymbol{\mu}^1, m)\Gamma(\boldsymbol{\mu}^1, \bar{\boldsymbol{\mu}}, m), \ldots, \tilde{W}(\boldsymbol{\mu}^{n_r}, m)\Gamma(\boldsymbol{\mu}^{n_r}, \bar{\boldsymbol{\mu}}, m)\right], \tag{24}$$

where the diagonal entries of $\Gamma(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}, m)$ are defined by Eq. (23). The POD basis can then be computed using this weighted snapshot matrix in algorithm 1. Unfortunately, this approach can lead to excessive weighting of snapshots computed at configurations "far" from the target configuration.

To demonstrate this, consider a problem with a single parameter ($n_\mu = 1$) and two reference configurations ($n_r = 2$). If, for example, $\mu^1 = 0.1\bar{\mu}$ and $\mu^2 = 0.8\bar{\mu}$, the weighted snapshot matrix of Eq. (24) would be

$$W = \left[u\left(\mu^1\right), 0.9\bar{\mu}\frac{du}{d\mu}\left(\mu^1\right), u\left(\mu^2\right), 0.2\bar{\mu}\frac{du}{d\mu}\left(\mu^2\right)\right]. \tag{25}$$

Even though the second configuration is closer to the target configuration in the parameter space $\mathcal{D}$, its sensitivity snapshot is given a smaller weight than that of the more distant configuration. As a result, the truncated POD basis computed using this matrix in algorithm 1 better represents the snapshots from the more distant (and therefore less relevant) configuration.

To remedy this, we introduce a *configuration distance weight*. Since $\mathcal{D} \subset \mathbb{R}^{n_\mu}$ we can use the Euclidean ℓ_2 norm to define distances between configurations.[c] To give smaller weights to snapshots computed at more distant points, we define the configuration distance weight as

$$\gamma^{\mathcal{D}}\left(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}\right) \equiv \|\bar{\boldsymbol{\mu}} - \boldsymbol{\mu}^i\|_2^{-\omega}, \tag{26}$$

where $\omega \in \mathbb{R}^+$ is a constant determined from numerical experiments.

We now define the Compact POD snapshot matrix of order m W_{C^m} as

$$W_{C^m}\left(\mathcal{D}_r, \bar{\boldsymbol{\mu}}\right) \equiv \left[\tilde{W}(\boldsymbol{\mu}^1, m)\Gamma_{C^m}(\boldsymbol{\mu}^1, \bar{\boldsymbol{\mu}}), \ldots, \tilde{W}(\boldsymbol{\mu}^{n_r}, m)\Gamma_{C^m}(\boldsymbol{\mu}^{n_r}, \bar{\boldsymbol{\mu}})\right] \tag{27}$$

$$\Gamma_{C^m}(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}) \equiv \gamma^{\mathcal{D}}\left(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}\right) \mathrm{diag}\left(\gamma_j^{\mathcal{T}}\left(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}\right)\right), \tag{28}$$

where $\mathcal{D}_r = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^{n_r}\} \subset \mathcal{D}$ is the set of reference configurations. To illustrate the utility of the configuration distance weights, we apply Eqs. (27)–(28) to the previous example with $\omega = 2$ in Eq. (26). This results in the following Compact POD snapshot matrix:

$$W_{C^1}\left(\{\mu^1, \mu^2\}, \bar{\mu}\right) = \left[\frac{1.23}{(\bar{\mu})^2}u\left(\mu^1\right), \frac{1.11}{\bar{\mu}}\frac{du}{d\mu}\left(\mu^1\right), \frac{25}{(\bar{\mu})^2}u\left(\mu^2\right), \frac{5}{\bar{\mu}}\frac{du}{d\mu}\left(\mu^2\right)\right]. \tag{29}$$

Since the second configuration is closer to the target, it is apparent that the distance weights cause CPOD to more heavily weight snapshots from nearer configurations as desired.

Thus, the matrix $W_{C^m}\left(\mathcal{D}_r, \bar{\boldsymbol{\mu}}\right)$ can be used in algorithm 1 to compute the *Compact POD basis of order* m, which we denote by Φ_{C^m}. This basis is optimal in the sense of the $(m+1)$-order Taylor reconstruction of the solution at the target with a correction made for configuration distances.

We now outline some of the advantages of CPOD. As noted previously, one shortcoming of reduced basis approximations is that the reduced basis is not generally robust with respect to parameter changes. However, since the CPOD basis is constructed from the sensitivities, which directly represent changes in the solution due to parameter variations, it is naturally more robust in the parameter space. Furthermore, the CPOD basis is optimal with respect to the reconstructed solution at a target configuration, whereas the typical POD basis is optimal only at the reference configuration(s). Thus, we expect CPOD to provide a more accurate approximate solution at the target than standard POD.

Furthermore, CPOD extracts more information at each reference configuration than the typical approach.

[c] This distance measure is meaningful only if the parameters are similarly scaled.

Thus, we expect CPOD to require many fewer reference configurations to obtain a desired level of accuracy in the approximate solution. A similar effect is observed when compact differencing schemes are used to approximate spatial derivatives.

We also note that the CPOD basis of order zero with $\omega = 0$ in Eq. (26) is equivalent to the typical POD basis. Thus, the standard POD method can be considered a special case of CPOD.

III.D. Computational cost savings

Beyond the theoretical appeal of the Compact POD method, there is also a practical advantage in terms of computational cost. Namely, the sensitivities are often inexpensive to compute because fast multiple-RHS techniques can be used to solve the (discrete, analytical) sensitivity equations.

Again, we consider the static system Eq. (5). If $\mathcal{R}$ represents a nonlinear operator, we obtain the following first- and second-order sensitivity equations by enforcing Eq. (5) for arbitrary perturbations in the parameters:

$$\frac{\partial \mathcal{R}}{\partial u}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_j}\left(\boldsymbol{\mu}\right) = -\frac{\partial \mathcal{R}}{\partial \mu_j}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right), \quad 1 \le j \le n_\mu \tag{30}$$

$$\begin{aligned}\frac{\partial \mathcal{R}}{\partial u}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)\frac{\partial^2 u}{\partial \mu_k \partial \mu_l}\left(\boldsymbol{\mu}\right) = &-\frac{\partial^2 \mathcal{R}}{\partial \mu_k \partial \mu_l}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) - \frac{\partial^2 \mathcal{R}}{\partial u \partial \mu_k}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_l}\left(\boldsymbol{\mu}\right) \\ &- \frac{\partial^2 \mathcal{R}}{\partial u \partial \mu_l}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_k}\left(\boldsymbol{\mu}\right), \quad 1 \le k,l \le n_\mu\end{aligned} \tag{31}$$

Eqs. (30)–(31) can be solved by multiple-RHS methods because the matrix $\frac{\partial \mathcal{R}}{\partial u}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right)$ is independent of parameter indices j, k, and l. Thus, in order to compute the sensitivities $\frac{\partial u}{\partial \mu_j}$, $1 \le j \le n_\mu$ and $\frac{\partial^2 u}{\partial \mu_k \partial \mu_l}$, $1 \le k \le l \le n_\mu$, we must only compute the "pseudo-loads" appearing on the RHS and solve repeatedly with the same matrix.

For linear $\mathcal{R}$, the sensitivity equations are

$$\frac{\partial \mathcal{R}}{\partial u}\left(\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_j}\left(\boldsymbol{\mu}\right) = -\frac{\partial \mathcal{R}}{\partial \mu_j}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right), \quad 1 \le j \le n_\mu \tag{32}$$

$$\begin{aligned}\frac{\partial \mathcal{R}}{\partial u}\left(\boldsymbol{\mu}\right)\frac{\partial^2 u}{\partial \mu_k \partial \mu_l}\left(\boldsymbol{\mu}\right) = &-\frac{\partial^2 \mathcal{R}}{\partial \mu_k \partial \mu_l}\left(u\left(\boldsymbol{\mu}\right);\boldsymbol{\mu}\right) - \frac{\partial^2 \mathcal{R}}{\partial u \partial \mu_k}\left(\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_l}\left(\boldsymbol{\mu}\right) \\ &- \frac{\partial^2 \mathcal{R}}{\partial u \partial \mu_l}\left(\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_k}\left(\boldsymbol{\mu}\right), \quad 1 \le k,l \le n_\mu\end{aligned} \tag{33}$$

Again, we can solve Eqs. (32)–(33) by multiple-RHS methods. Note that we can also compute even higher-order sensitivities using multiple-RHS techniques, since the same matrix appears on the LHS for sensitivity equations of all orders.

The best multiple-RHS method to use depends on the bandedness of the matrix at hand. For matrices with small bandwidth, the matrix factorization can be computed once and then be reused for each RHS by performing forward- and back-substitution. For systems not meeting this criteria, multiple-RHS methods based on the reuse of Krylov subspaces[33] can be used. Here, the iterative method's objective function is first (cheaply) minimized in the Krylov subspace $\mathcal{K}$ generated from solving previous systems. Next, the function is minimized over the complement of this subspace by enforcing orthogonality of the search directions to $\mathcal{K}$. As a result, solving the linear system with each additional RHS is computationally less expensive than solving the original system.

Thus, Compact POD with multiple-RHS solution methods provides an advantage over the typical POD approach in two settings. For a fixed number of reference configurations n_r, we can compute $n_C(m) - 1$ *more* snapshots at a slightly increased cost. If we instead desire a specific number of snapshots n_s, we can compute them at a *reduced* cost using fewer reference configurations and multiple-RHS methods. This advantage is particularly dramatic when applied to nonlinear static systems, since solving for the solution itself

requires a nonlinear system solution strategy such as Newton's method, whereas solving for the sensitivity requires the solution of only a single linear system (with a multiple RHS method).

Finally, we note that if the cost of computing the pseudo-loads[34] appearing on the RHS of Eqs. (30)–(33) overrides the savings achieved by multiple-RHS methods, we may revert to finite difference methods to compute the sensitivities. In this case, the cost of CPOD is comparable to that of the typical POD approach if low-order differences (e.g. one-sided difference for first derivatives, three-point centered difference for second derivatives) are used. Thus, computing snapshots for CPOD is at least as inexpensive as doing so using the typical approach.

III.E. Integration with optimization and adaptive sampling

We now turn to the integration of the Compact POD method with gradient-based optimization algorithms. As mentioned in Section I, the optimization of static systems has been limited to offline-online decomposition procedures, where the dominant cost is incurred in the offline snapshot generation phase. By incorporating Compact POD within this framework, we can decrease the cost of the offline phase by exploiting multiple-RHS solution methods (see Section III.D).

However, we would like to avoid offline-online procedures if possible due to its innate drawbacks. These include the aforementioned 'break-even' point and the lack of correlation between the reference configurations and the optimization trajectory (see figure 1). As previously discussed, strategies that employ adaptive sampling and basis updating techniques avoid these shortcomings, but require the possibility of computing multiple snapshots at a single configuration for fixed boundary conditions. Since CPOD enables this for static systems, we can now use these adaptive methodologies in this context.

As an example of this, we consider an extension of OS-POD[6] to the optimization of static systems via the CPOD basis. Consider the following design optimization problem:

$$\begin{aligned} \underset{u,\boldsymbol{\mu}}{\text{minimize}} \quad & J\left(u,\boldsymbol{\mu}\right) \\ \text{subject to} \quad & c_i(u,\boldsymbol{\mu}) = 0, \quad 1 \le i \le n_{ec} \\ & d_j\left(u,\boldsymbol{\mu}\right) \ge 0, \quad 1 \le j \le n_{ic} \\ & \mathcal{R}\left(u;\boldsymbol{\mu}\right) = 0. \end{aligned} \tag{34}$$

Here, $J : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}$ is the objective function to indicate system performance, $c_i : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}$ and $d_j : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}$ represent constraints, and $\mathcal{R}$ is associated with the governing equations (5). If the objective function and constraints are continuously differentiable in u and $\boldsymbol{\mu}$, we can use any of a variety of gradient-based optimization algorithms[35] to solve problem (34).

We proceed by introducing reduced basis approximation in problem (34):

$$\begin{aligned} \underset{\tilde{u},\boldsymbol{\mu}}{\text{minimize}} \quad & J\left(\tilde{u},\boldsymbol{\mu}\right) \\ \text{subject to} \quad & c_i(\tilde{u},\boldsymbol{\mu}) = 0, \quad 1 \le i \le n_{ec} \\ & d_j\left(\tilde{u},\boldsymbol{\mu}\right) \ge 0, \quad 1 \le j \le n_{ic} \\ & \\ & \Phi^T \mathcal{R}\left(\Phi\hat{u};\boldsymbol{\mu}\right) = 0 \\ & \tilde{u} = \Phi\hat{u}. \end{aligned} \tag{35}$$

Problem (35) does not specify how Φ is computed. The direct extension of OS-POD to static systems would choose Φ to be the (typical) POD basis using only the full-order solution $u(\boldsymbol{\mu})$ as a snapshot. However, this would produce a reduced basis of dimension one, which is too small to be practical.

Instead, we choose the CPOD basis of order one, which results in the following problem:

$$
\begin{aligned}
\underset{\tilde{u},\boldsymbol{\mu}}{\text{minimize}} \quad & J\left(\tilde{u},\boldsymbol{\mu}\right) \\
\text{subject to} \quad & c_i(\tilde{u},\boldsymbol{\mu}) = 0, \quad 1 \le i \le n_{ec} \\
& d_j\left(\tilde{u},\boldsymbol{\mu}\right) \ge 0, \quad 1 \le j \le n_{ic} \\
& \\
& \Phi^T \mathcal{R}\left(\Phi\hat{u};\boldsymbol{\mu}\right) = 0 \\
& \tilde{u} = \Phi\hat{u} \\
& \\
& \mathcal{R}\left(u;\boldsymbol{\mu}\right) = 0 \\
& \frac{\partial \mathcal{R}}{\partial u}\left(u;\boldsymbol{\mu}\right)\frac{\partial u}{\partial \mu_j} = -\frac{\partial \mathcal{R}}{\partial \mu_j}\left(u;\boldsymbol{\mu}\right), \quad 1 \le j \le n_\mu \\
& \Phi = \Phi_{\,C^1}\left(n_{\phi,\mathcal{E}}\right) \text{ by } W_{C^1}\left(\boldsymbol{\mu},\boldsymbol{\mu}\right) \text{ in algorithm 1, with } \omega = 0 \text{ in Eq. (26).}
\end{aligned}
\tag{36}
$$

Algorithm 2 can be used to solve problem (36). The figure contains a graphical representation of this implementation.

Algorithm 2 Extension of OS-POD to static systems with CPOD.

1: $n \leftarrow 1$ and choose initial parameters $\boldsymbol{\mu}^1 \in \mathcal{D}$.
2: Compute $u(\boldsymbol{\mu}^n)$, $\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu}^n)$, $1 \le j \le n_\mu$.
3: Use $u(\boldsymbol{\mu}^n)$, $\frac{\partial u}{\partial \mu_j}(\boldsymbol{\mu}^n)$, $1 \le j \le n_\mu$ in a single gradient step of problem (34) to compute intermediate parameters $\bar{\boldsymbol{\mu}}$.
4: Compute $u(\bar{\boldsymbol{\mu}})$. If $u(\bar{\boldsymbol{\mu}})$ and $\bar{\boldsymbol{\mu}}$ satisfy optimality conditions for problem (34), stop.
5: Define $\hat{W}_{C^1}(\{\boldsymbol{\mu}^n,\bar{\boldsymbol{\mu}}\},\bar{\boldsymbol{\mu}}) \equiv [W_{C^1}(\boldsymbol{\mu}^n,\bar{\boldsymbol{\mu}}), u(\bar{\boldsymbol{\mu}})]$ where W_{C^1} is computed with $\omega = 0$ in Eq. (26). Compute $\Phi_{C^1}(n_{\phi,\mathcal{E}})$ by using $\hat{W}_{C^1}(\{\boldsymbol{\mu}^n,\bar{\boldsymbol{\mu}}\},\bar{\boldsymbol{\mu}})$ in algorithm 1.
6: Solve problem (35) with $\Phi = \Phi_{C^1}(n_{\phi,\mathcal{E}})$ using a gradient-based optimization algorithm.
7: $n \leftarrow n+1$, return to step 2.

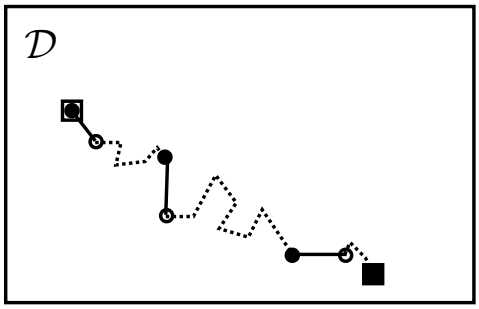


Figure 2. Progression of Algorithm 2. $\mathcal{D}$ parameter space, — optimization trajectory (FOM), ··· optimization trajectory (ROM), □ initial parameters, ■ optimal parameters, • sample parameters (solution and sensitivities computed), ∘ sample parameters (only solution computed).

This algorithm illustrates a major advantage of using the CPOD basis in adaptive sampling strategies for optimization: the full-order solutions serve a dual purpose. Namely, these solutions both advance the optimization trajectory and are used to compute the reduced basis. This is apparent in algorithm 2, where the full-order solutions computed in step 2 advance the convergence of the optimal solution via the gradient step in step 3 and are used again to compute the CPOD basis in step 5. This is in contrast to OS-POD as presented in Ref. [6], where the full-order solutions computed at $\boldsymbol{\mu}^n$ are only used to calculate a gradient step for problem (34) and do not contribute to the computation of the reduced basis. Thus, the inclusion of the CPOD basis in the proposed extension of this method allows us to more fully exploit the expensive full-order computations.

We also note that when $n_{ec} + n_{ic} + 1 > n_\mu$, the direct method should be used for sensitivity analysis when solving problem (34). In this case, problem (36) is a particularly well-suited surrogate problem, because steps 2–4 of algorithm 2 are identical to steps taken in any implementation to solve problem (34).

IV. Example

Here, we apply the Compact POD basis to the analysis of a parameterized plate in plane stress to compare its quality with that of the standard POD basis.

IV.A. Finite element model

We consider a parameterized, rectangular, cantilevered plate in plane stress with a uniformly distributed load. The base configuration with $\mu_i = 0$, $1 \leq i \leq n_\mu$ is shown in figure 3. The base domain is of (dimensionless) size $L = 40$, $h = 20$, and in-plane thickness $t = 0.11$. The base material of the plate is characterized by the modulus of elasticity $E = 1 \times 10^5$, Poisson's ratio $\nu = 0.3$, and material density $\rho = 1.0$. The total magnitude of the distributed load is 1000.

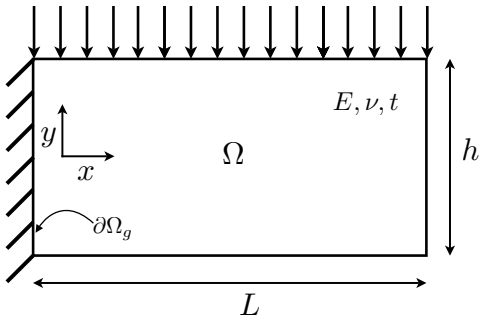


Figure 3. Domain of base configuration.

The standard weak form of the problem describing the static displacement field of the parameterized structure with homogeneous Dirichlet boundary conditions is: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathsf{u} \in \mathcal{S}\big(\Omega(\boldsymbol{\mu})\big)$ satisfying

$$a(\mathsf{u}, \mathsf{v}; \boldsymbol{\mu}) = f(\mathsf{v}), \quad \forall \mathsf{v} \in \mathcal{S}. \tag{37}$$

Here, the space of trial functions is

$$\mathcal{S}\big(\Omega(\boldsymbol{\mu})\big) \equiv \{\mathsf{u} \in \Big(H^1\big(\Omega(\boldsymbol{\mu})\big)\Big)^2 \mid \mathsf{u}|_{\partial\Omega_g} = 0\}, \tag{38}$$

where $H^1\big(\Omega(\boldsymbol{\mu})\big)$ is the Hilbert space

$$H^1(\Omega) \equiv \{\mathsf{u} \in L^2(\Omega) \mid \nabla\mathsf{u} \in \big(L^2(\Omega)\big)^2\}, \tag{39}$$

and $L^2(\Omega)$ is the space of square integrable functions over the domain. $a(\cdot, \cdot; \boldsymbol{\mu}) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a symmetric, bilinear, coercive form and represents the stiffness of the plate. The linear form $f : \mathcal{S} \to \mathbb{R}$ characterizes the (parameter-independent) distributed load. Due to the properties of $a(\cdot, \cdot; \boldsymbol{\mu})$, we can define the standard energy inner product and norm as $(\mathsf{u}, \mathsf{v})_{E,\boldsymbol{\mu}} \equiv a(\mathsf{u}, \mathsf{v}; \boldsymbol{\mu})$ and $\|\mathsf{u}\|_{E,\boldsymbol{\mu}} \equiv \sqrt{a(\mathsf{u}, \mathsf{u}; \boldsymbol{\mu})}$ respectively.

We introduce the finite element discretization characterized by a finite set of basis functions $\xi_i \in \mathcal{S}$, $1 \leq i \leq N$ and subspace $\mathcal{S}^h \equiv \text{span}\{\xi_i, 1 \leq i \leq N\} \subset \mathcal{S}$. When we choose to solve Eq. (37) over $\mathcal{S}^h$, the problem becomes: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathsf{u}^h \in \mathcal{S}^h\big(\Omega(\boldsymbol{\mu})\big)$ satisfying

$$a(\mathsf{u}^h, \mathsf{v}^h; \boldsymbol{\mu}) = f(\mathsf{v}^h), \quad \forall \mathsf{v}^h \in \mathcal{S}^h. \tag{40}$$

We implement this discretization via a mesh consisting of 2048 quadrilateral elements with linear shape functions, two degrees of freedom (translational displacements) per node, and unit aspect ratio. This results

in $N = 4224$ and a relative (approximate) error in the energy norm of the finite element solution u^h of 4.0%, which is accurate enough for our purposes.[d] The matrix-vector representation of the discrete weak form in Eq. (40) is

$$K(\boldsymbol{\mu})\, u(\boldsymbol{\mu}) = F, \tag{41}$$

where $K : \mathcal{D} \to \mathbb{R}^{N\times N}$ and $F \in \mathbb{R}^N$ are defined by $K_{ij}(\boldsymbol{\mu}) = a(\xi_i; \xi_j; \boldsymbol{\mu})$ and $F_i = f(\xi_i)$. The finite element solution on $\Omega(\boldsymbol{\mu})$ can be recovered from the state vector coordinates $u(\boldsymbol{\mu}) = [u_1(\boldsymbol{\mu}), \ldots, u_N(\boldsymbol{\mu})]^T \in \mathbb{R}^N$ by

$$\mathsf{u}^h\big(u(\boldsymbol{\mu})\big) = \sum_{i=1}^{N} u_i(\boldsymbol{\mu})\, \xi_i. \tag{42}$$

Note that we can equivalently write the energy inner product and norm in terms of the state vector as $(u, v)_{K(\boldsymbol{\mu})} = \big(\mathsf{u}(u), \mathsf{v}(v)\big)_{E,\boldsymbol{\mu}}$ and $\|u\|_{K(\boldsymbol{\mu})} = \|\mathsf{u}(u)\|_{E,\boldsymbol{\mu}}$, where

$$(u, v)_{K(\boldsymbol{\mu})} \equiv u^T K(\boldsymbol{\mu}) v \tag{43}$$

$$\|u\|_{K(\boldsymbol{\mu})} \equiv \sqrt{u^T K(\boldsymbol{\mu}) u}. \tag{44}$$

IV.B. Design model

We now parameterize the structure using eleven variables ($n_\mu = 11$) representing planform shape, thickness, and material property changes.

To implement the shape parameters, we use the geometric modeler SDESIGN, developed at the University of Colorado at Boulder by Maute and co-workers. This package uses the "design element" concept to smoothly deform a given mesh via control node displacements.[37] Thus, these displacements define the shape parameters. To create the design model shown in figure 4, we construct a design box (cuboid) around the finite element mesh and define eight control nodes.[e] We specify the top and bottom edges to deform as cubic Bézier splines, and the left and right edges to deform linearly based on control node displacements. Using

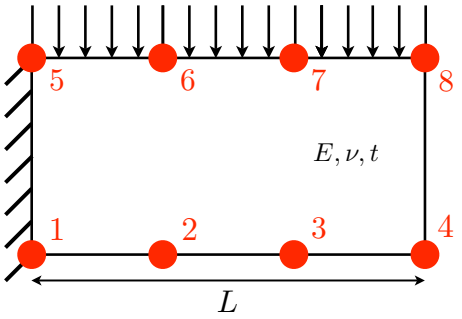


Figure 4. Design model. Control nodes shown in red.

this environment, we define nine shape parameters: the vertical displacement of each control node and the length L of the plate.

We also treat the thickness t and Poisson's ratio ν as parameters. The eleven physical parameters p_i, $1 \le i \le n_\mu$ and their bounds $p_{i,lb}$ and $p_{i,ub}$, $1 \le i \le n_\mu$ are listed in table 1. To ensure reasonable scaling, we define the system parameters from the physical parameters via the affine mapping

$$\mu_i(p_i) = \frac{2}{p_{i,ub} - p_{i,lb}} (p_i - p_{i,0}), \quad i = 1, \ldots, n_\mu. \tag{45}$$

Since $p_{i,0} = \frac{1}{2}(p_{i,ub} - p_{i,lb})$, we have $\mu_i \in [-1, 1]$, $i = 1, \ldots, n_\mu$. This gives $\mathcal{D} = [-1, 1]^{n_\mu}$.

[d] This was determined by $h-$extension using the procedure in Ref. [36].

[e] Although the design box was three-dimensional, it was constructed symmetrically and thus can be considered a 2-D plane.

i	1	2	3	4	5	6	7	8	9	10	11
Definition	Δy_1	Δy_2	Δy_3	Δy_4	Δy_5	Δy_6	Δy_7	Δy_8	L	t	ν
$p_{i,0}$	0	0	0	0	0	0	0	0	40	0.11	0.3
$p_{i,lb}$	-5	-5	-5	-5	-5	-5	-5	-5	20	0.06	0.2
$p_{i,ub}$	5	5	5	5	5	5	5	5	60	0.16	0.4

Table 1. Physical system parameters p_i.

IV.C. ROM construction via Galerkin projection

In order to construct a ROM for this problem, we use a basis $\Phi = \left[\phi_1, \ldots, \phi_{n_\phi}\right] \in \mathbb{R}^{N \times n_\phi}$ for the state vector u such that $\tilde{\mathcal{S}}\left(\Omega, \Phi\right) \equiv \text{span}\{\sum_{i=1}^{N} \xi_i \phi_{j,i}, 1 \leq j \leq n_\phi\} \subset \mathcal{S}^{\,h} \subset \mathcal{S}$ and project Eq. (40) onto $\tilde{\mathcal{S}}$ via Galerkin projection. This results in the following weak form: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\tilde{\mathsf{u}} \in \tilde{\mathcal{S}}\left(\Omega\left(\boldsymbol{\mu}\right), \Phi\right)$ satisfying

$$a(\tilde{\mathsf{u}}, \tilde{\mathsf{v}}; \boldsymbol{\mu}) = f\left(\tilde{\mathsf{v}}\right), \quad \forall \tilde{\mathsf{v}} \in \tilde{\mathcal{S}}. \tag{46}$$

This problem corresponds directly to projecting Eq. (41) in the following way:

$$\Phi^T K(\boldsymbol{\mu}) \Phi \hat{u} = \Phi^T F \tag{47}$$

$$\tilde{u}\left(\Phi, \boldsymbol{\mu}\right) = \Phi \hat{u}. \tag{48}$$

Note that Eqs. (47)–(48) correspond to the Galerkin projection of a general linear system as in Eqs. (8)–(9).

Since $\tilde{\mathcal{S}} \subset \mathcal{S}^h$, we can use Eq. (40) in Eq. (46) and the definition of the energy inner product to obtain

$$(\mathsf{u}^h - \tilde{\mathsf{u}}, \tilde{\mathsf{v}})_{E,\boldsymbol{\mu}} = 0, \quad \forall \tilde{\mathsf{v}} \in \tilde{\mathcal{S}}. \tag{49}$$

The corresponding discrete form is

$$(u - \tilde{u}, \phi_i)_{K(\boldsymbol{\mu})} = 0, \quad i = 1, \ldots, n_\phi, \tag{50}$$

where $\mathsf{u}^h = \mathsf{u}^h(u)$ and $\tilde{\mathsf{u}} = \tilde{\mathsf{u}}(\tilde{u})$ $\big($see Eq. (42)$\big)$. Since $\tilde{u} \in \text{span}\{\phi_i, 1 \leq i \leq n_\phi\}$, we can rewrite Eq. (12) for the error of the approximate solution in this space as

$$e_{\parallel} = \sum_{i=1}^{n_\phi} (u - \tilde{u}, \phi_i)_\Theta \, \phi_i. \tag{51}$$

By comparing Eqs. (50) and (51), we see that by choosing $\Theta = K(\boldsymbol{\mu})$, our approximate solution has $e_{\parallel} = 0$ and therefore $e = e_\perp$ due to this "best approximation property" of the finite element method in the energy norm. This choice is also amenable to the POD framework because the POD basis is constructed with the aim of minimizing the projection error $e_\perp$, which becomes the total error for $\Theta = K(\boldsymbol{\mu})$. Additionally, we have found that using $\Theta = K(\boldsymbol{\mu})$ leads to a better correlation between the retained energy in Eq. (3) and the accuracy of approximate solution than for the choice $\Theta = I$.[f] Therefore, we set $\Theta = K(\boldsymbol{\mu})$ for our numerical experiments.

IV.D. Numerical procedure

We now present the numerical procedure to test the validity of the Compact POD method. We compare the approximate solutions obtained by solving Eqs. (47)–(48) with three different bases: the CPOD basis of order one (Φ_{C^1}), the typical POD basis using both the same number of reference configurations ($\Phi_{\text{POD},r}$) as CPOD, and the typical POD basis employing the same number of snapshots ($\Phi_{\text{POD},s}$) as CPOD.

We first define a set of n_r randomly-chosen reference configurations $\mathcal{D}_r = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^{n_r}\} \subset \mathcal{D}$ at which we compute the solution and sensitivities. We then form unweighted snapshot matrices for the CPOD basis $\tilde{W}(\boldsymbol{\mu}, 1)$, $\forall \boldsymbol{\mu} \in \mathcal{D}_r$ via Eq. (17). Next, we compute the untruncated basis $\Phi_{\text{POD},r}$ by using $W_{\text{POD},r} =$

[f] Similar results were found in Ref. [38], where the $H^1(\Omega)$ inner product, which utilizes higher-order information like the energy norm, provided a better correlation than the $L^2(\Omega)$ inner product.

$\left[u\left(\boldsymbol{\mu}^1\right), \ldots, u\left(\boldsymbol{\mu}^{n_r}\right)\right]$ in steps 1–3 of algorithm 1. Note that this results in $n_r n_C(1) = n_r\left(n_\mu + 1\right)$ snapshots for the CPOD approach, and n_r snapshots for the POD approach using the same n_r configurations.

To compute $\Phi_{\mathrm{POD},s}$, we require $n_r n_\mu$ more reference configurations. We therefore define an additional set of configurations $\tilde{\mathcal{D}}_r = \{\boldsymbol{\mu}^{n_r+1}, \ldots, \boldsymbol{\mu}^{n_r(n_\mu+1)}\} \subset \mathcal{D}$ at which we compute the solution. We then set $W_{\mathrm{POD},s} = \left[u(\boldsymbol{\mu}^1), \ldots, u(\boldsymbol{\mu}^{n_r(n_\mu+1)})\right]$ and use it in steps 1–3 of algorithm 1 to compute the untruncated basis $\Phi_{\mathrm{POD},s}$.

We next choose a random target configuration defined by $\bar{\boldsymbol{\mu}} \in \mathcal{D}$ at which we compare the various approximate solutions. Then, we compute the untruncated CPOD basis by first calculating $W_{C^1}(\mathcal{D}_r, \bar{\boldsymbol{\mu}})$ via Eq. (27), and then employing steps 1–3 of algorithm 1. We determined $\omega = 6$ to be appropriate, and thus we use it in Eq. (26) to compute the weighted snapshot matrix.

To compare all three methods, we require a uniform truncation of the bases to ensure that the reduced-order models are of the same dimension. Thus, we apply steps 4–5 of algorithm 1 to all three bases using $n_\phi = n_r$, since this is the maximum allowable dimension for $\Phi_{\mathrm{POD},r}$. We next compute approximate solutions $\tilde{u}\left(\Phi_{\mathrm{POD},r}\left(n_r\right), \bar{\boldsymbol{\mu}}\right)$, $\tilde{u}\left(\Phi_{\mathrm{POD},s}\left(n_r\right), \bar{\boldsymbol{\mu}}\right)$, and $\tilde{u}\left(\Phi_{C^1}\left(n_r\right), \bar{\boldsymbol{\mu}}\right)$ and their corresponding errors.

To compare Φ_{C^1} and $\Phi_{\mathrm{POD},s}$ using a ROM of more reasonable dimension, we compute $n_{\phi,\mathcal{E}}\left(\bar{\Sigma}_{C^1}, 0.9999\right)$ via Eq. (4), where $\bar{\Sigma}_{C^1}$ are the singular values corresponding to the Compact POD snapshot matrix. We truncate the bases using steps 4–5 of algorithm 1 with $n_\phi = n_{\phi,\mathcal{E}}$ and then compute $\tilde{u}\left(\Phi_{\mathrm{POD},s}\left(n_{\phi,\mathcal{E}}\right), \bar{\boldsymbol{\mu}}\right)$ and $\tilde{u}\left(\Phi_{C^1}\left(n_{\phi,\mathcal{E}}\right), \bar{\boldsymbol{\mu}}\right)$ as well as the corresponding errors.

We do not compare these methods with the Hermite subspace approach discussed in Section III.A because there is no mechanism to truncate its basis. We only note that the Hermite subspace is equivalent to the span of the untruncated CPOD basis of order one ($\mathcal{H} = \mathrm{span}\{\phi_{C^1 i}, 1 \leq i \leq n_s\}$).

IV.E. Reference solution proximity measures

Because the Compact POD method computes multiple snapshots at each reference configuration, Φ_{C^1} and $\Phi_{\mathrm{POD},s}$ must use different sets of reference configurations ($\mathcal{D}_r$ and $\mathcal{D}_r \cup \tilde{\mathcal{D}}_r$, respectively) to collect the same number of total snapshots. So, we expect the relative qualities of their approximate solutions to depend in some way on the relative "nearness" of their reference configurations to the target configuration.

We therefore introduce several proximity metrics. Since we are using the finite element method with $\Theta = K(\boldsymbol{\mu})$, we have $e = e_\perp$ (see Section IV.C). So, a reference configuration can be considered close to the target if the following relative projection error is small:

$$\begin{aligned} \delta\left(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}\right) &\equiv \frac{\| e_\perp\left(\hat{u}\left(\boldsymbol{\mu}^i\right), \bar{\boldsymbol{\mu}}\right) \|_{K(\bar{\boldsymbol{\mu}})}}{\| u\left(\bar{\boldsymbol{\mu}}\right) \|_{K(\bar{\boldsymbol{\mu}})}} \\ \mathrm{span}\{\hat{u}\left(\boldsymbol{\mu}^i\right)\} &= \mathrm{span}\{u\left(\boldsymbol{\mu}^i\right)\} \\ \| \hat{u}\left(\boldsymbol{\mu}^i\right) \|_{K(\bar{\boldsymbol{\mu}})} &= 1. \end{aligned} \tag{52}$$

This indicates the relative projection error of the full-order solution at the target configuration onto the subspace spanned by the solution at the i^{th} reference configuration. We specifically consider the mean and minimum of this quantity over a set of reference configurations $\mathcal{D}_a = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^{n_a}\}$:

$$\bar{\delta}\left(\mathcal{D}_a, \bar{\boldsymbol{\mu}}\right) = \frac{1}{n_a} \sum_{i=1}^{n_a} \delta(\boldsymbol{\mu}^i, \bar{\boldsymbol{\mu}}) \tag{53}$$

$$\delta_{\min}\left(\mathcal{D}_a, \bar{\boldsymbol{\mu}}\right) = \min_{\boldsymbol{\mu} \in \mathcal{D}_a \subset D} \delta\left(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}\right). \tag{54}$$

These metrics lend insight to the advantage of each approach due to the quality of data it employs.

IV.F. Results for five reference configurations

We begin by determining the approximate costs of computing the snapshots for each method as measured in computational time. Due to the moderate size and small bandwidth of K, we use a direct method to solve Eq. (41) for various $u(\boldsymbol{\mu})$ in the snapshot collection phase. To compute the sensitivities, we use a direct multiple-RHS method. That is, we store the Cholesky factors of the stiffness matrix and perform a forward- and back-substitutution for each pseudo-load of Eq. (32). Table 2 indicates that, as expected, it is cheaper

Table 2. Computational time for snapshot collection (seconds), $n_r = 5$.

W_{C^1}	$W_{\text{POD},s}$	$W_{\text{POD},r}$
8.38	9.05	0.75

to generate a given number of snapshots for the CPOD method compared with the typical POD approach. Of course, for a given number of reference configurations, it is more expensive to compute the additional snapshots corresponding to the sensitivities due to the calculation of the pseudo-loads.

To compare performance, we consider the approximate solutions obtained at three different randomly-chosen target configurations. The distance metrics relating the reference configurations to these target configurations are provided in table 3. This table indicates that the reference data associated with both $\mathcal{D}_r$ and $\tilde{\mathcal{D}}_r$

Table 3. Reference solution distance measures, $n_r = 5$.

$\bar{\boldsymbol{\mu}}$	$\bar{\delta}(\mathcal{D}_r, \bar{\boldsymbol{\mu}})$	$\bar{\delta}(\mathcal{D}_r \cup \tilde{\mathcal{D}}_r, \bar{\boldsymbol{\mu}})$	$\delta_{\min}(\mathcal{D}_r, \bar{\boldsymbol{\mu}})$	$\delta_{\min}(\mathcal{D}_r \cup \tilde{\mathcal{D}}_r, \bar{\boldsymbol{\mu}})$
$\bar{\boldsymbol{\mu}}^1$	0.568	0.595	0.439	0.329
$\bar{\boldsymbol{\mu}}^2$	0.598	0.654	0.321	0.318
$\bar{\boldsymbol{\mu}}^3$	0.681	0.689	0.544	0.358

are quite distant from that of the target configuration. In fact, of the three target configurations, the closest reference solution still resulted in a projection error over 30% in the sense of Eq. (52). Thus, if we obtain an accurate solution with any of the bases, we can conclude that the POD method is reasonably robust for this problem. We also note that $\delta_{\min}$ is (as expected) smaller for the larger set. This implies that $\Phi_{\text{POD},s}$ uses more relevant data in the sense of this metric, which gives it an inherent advantage to produce an accurate solution. Of course, we anticipate the higher-order information exploited by the CPOD basis will offset this advantage to some degree.

The relative error in the energy norm between the full-order and approximate solutions is used to measure accuracy:

$$e_r\left(\Phi, \bar{\boldsymbol{\mu}}\right) = \frac{\|u\left(\bar{\boldsymbol{\mu}}\right) - \tilde{u}\left(\Phi, \bar{\boldsymbol{\mu}}\right)\|_{K(\bar{\boldsymbol{\mu}})}}{\|u\left(\bar{\boldsymbol{\mu}}\right)\|_{K(\bar{\boldsymbol{\mu}})}}. \tag{55}$$

Figure 5 contains the relative errors of the solution of all three methods as a function of ROM dimension n_ϕ. Note that the data related to $\Phi_{\text{POD},r}$ terminate at $n_\phi = n_r$ because the method is limited to computing only a single snapshot per configuration. We also note that the error related to the CPOD tends to drop rather quickly for small n_ϕ compared to the typical POD method. This implies that our weighting scheme in Eqs. (27)–(28) allows for accuracy to be retained even when the basis is significantly truncated as desired. We now consider results for some specific choices of ROM dimension n_ϕ.

Table 4 contains the relative errors computed by the three reduced basis methods for $n_\phi = n_r$. This table shows that Φ_{C^1} and $\Phi_{\text{POD},s}$ perform similarly, and their relative accuracies depend on the target configuration. However, both methods give more accurate results than $\Phi_{\text{POD},r}$ with only one exception. This implies that accuracy can generally be improved by computing an "excessive" number of snapshots ($n_s > n_\phi$) and truncating the basis. Furthermore, we note that $e_r\left(\Phi_{C^1}, \bar{\boldsymbol{\mu}}^i\right) < e_r\left(\Phi_{\text{POD},r}, \bar{\boldsymbol{\mu}}^i\right)\ \forall i$. We therefore conclude that the CPOD basis computes a much more accurate solution than the typical POD method when the same reference configurations and ROM dimension are used.

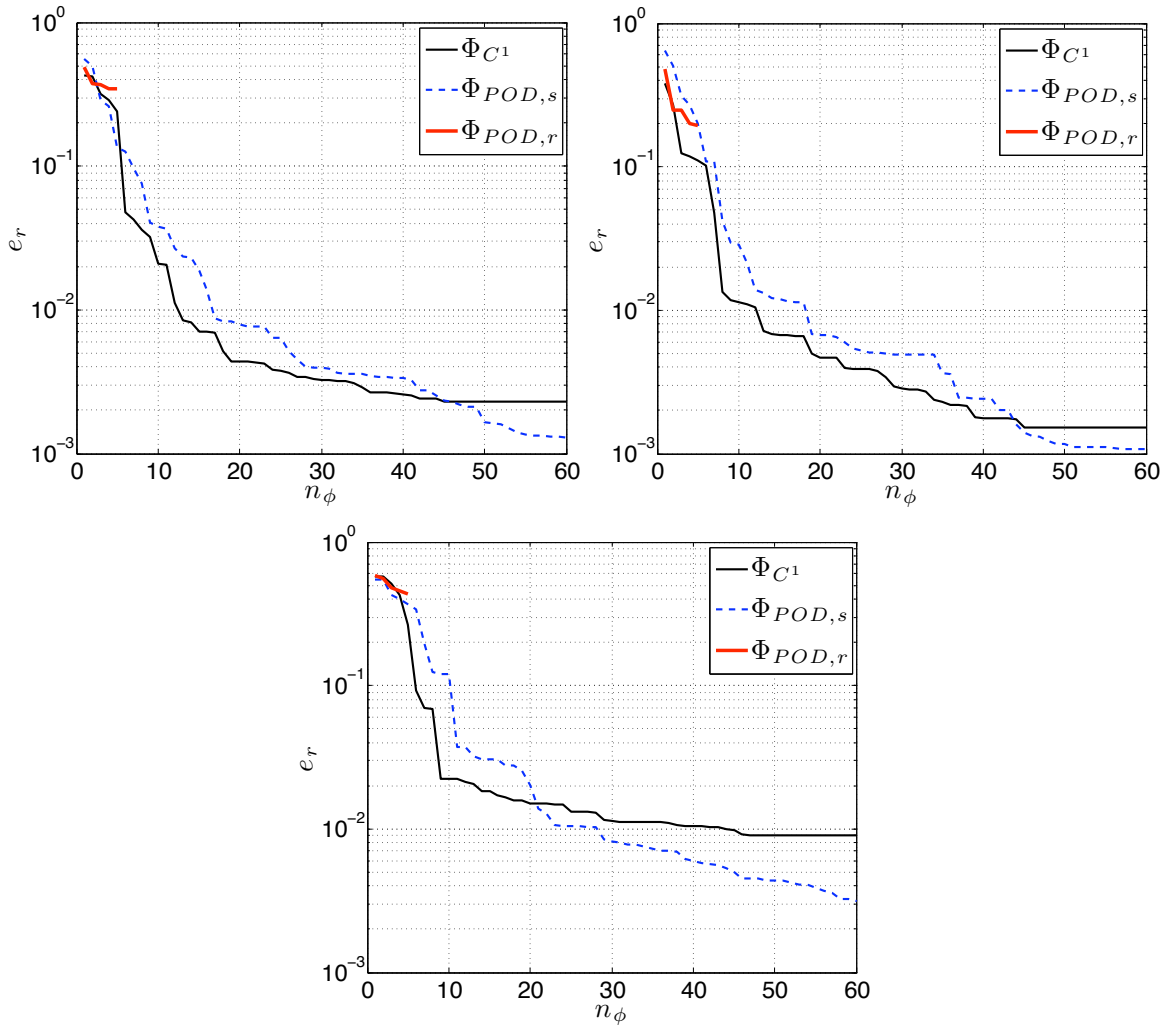


Figure 5. Dependence of relative error on n_ϕ, $n_r = 5$ for targets $\bar{\boldsymbol{\mu}}^1$, $\bar{\boldsymbol{\mu}}^2$, $\bar{\boldsymbol{\mu}}^3$ (clockwise from top left).

Table 4. Relative errors, $n_\phi = 5$, $n_r = 5$.

$\bar{\boldsymbol{\mu}}$	$e_r(\Phi_{C^1}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\mathrm{POD},s}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\mathrm{POD},r}, \bar{\boldsymbol{\mu}})$
$\bar{\boldsymbol{\mu}}^1$	0.2395	0.1351	0.3473
$\bar{\boldsymbol{\mu}}^2$	0.1112	0.2012	0.1938
$\bar{\boldsymbol{\mu}}^3$	0.2634	0.3701	0.4385

We now use the energy criterion Eq. (4) to compute $n_{\phi,\mathcal{E}}\left(\bar{\Sigma}_{C^1}, 0.9999\right)$ and obtain approximate solutions using a ROM of this dimension ($n_\phi = n_{\phi,\mathcal{E}}$). The results are provided in table 5. These data once again

Table 5. Relative errors, $n_\phi = n_{\phi,\mathcal{E}}$, $n_r = 5$.

$\bar{\boldsymbol{\mu}}$	$e_r(\Phi_{C^1}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\mathrm{POD},s}, \bar{\boldsymbol{\mu}})$	$n_{\phi,\mathcal{E}}$
$\bar{\boldsymbol{\mu}}^1$	0.71×10^{-2}	1.85×10^{-2}	15
$\bar{\boldsymbol{\mu}}^2$	1.17×10^{-2}	2.98×10^{-2}	9
$\bar{\boldsymbol{\mu}}^3$	1.84×10^{-2}	3.06×10^{-2}	14

demonstrate the similar performance of the Compact POD basis and typical POD basis utilizing the same number of snapshots. Also, we draw some interesting conclusions by referring to tables 2 and 3. Namely, for a fixed number of snapshots, the CPOD basis computes an approximate solution of similar or better accuracy than the standard POD method at a lower computational cost. Moreover, it does so using one twelfth as many reference configurations that are less relevant to the target (as measured by $\delta_{\min}$). Finally, we observe striking results for CPOD: using only five points in an 11-dimensional parameter space, it generates approximate solutions to within 2% of the full-order solution using a model reduced to less than 0.4% of the full-order description.

IV.G. Results for ten reference configurations

We now run the same numerical experiments using twice as many snapshots (note that the subsets represented by $\mathcal{D}_r$ and $\tilde{\mathcal{D}}_r$ have also doubled in size). The results are shown in figure 6 and tables 6–9. We observe the same general trends as before. We are still using distant reference configurations to compute our approximate solutions, as indicated by table 7. Also, we again observe a rapid decrease in the relative error for the Compact POD method in figure 6, which verifies the merit of the proposed weighting scheme. Again, when we fix the number of reference configurations and ROM dimension, table 8 shows that CPOD computes a much more accurate solution than the typical POD method.

Tables 8 and 9 demonstrate that the CPOD solution is more accurate than that of the standard POD method for all considered target configurations when a fixed number of snapshots is considered. Furthermore, this is accomplished with one twelfth as many reference configurations, at a reduced cost, and using less relevant data in the sense of $\delta_{\min}$. In fact, for the first target configuration, the CPOD method uses data that are less relevant in both the $\bar{\delta}$ and $\delta_{\min}$ measures. Lastly, we again note that CPOD produces excellent results in its own right: it computes approximate solutions with errors less than 1.5% using data at only 10 points in an 11-dimensional space, when the size of the model is drastically reduced to less than 0.4% the size of the full-order model. These results showcase the accuracy, efficiency, and robustness of the CPOD method.

Table 6. Computational time for snapshot collection (seconds), $n_r = 10$.

W_{C^1}	$W_{\mathrm{POD},r}$	$W_{\mathrm{POD},s}$
16.76	18.10	1.51

Table 7. Reference solution distance measures, $n_r = 10$.

$\bar{\boldsymbol{\mu}}$	$\bar{\delta}(\mathcal{D}_r, \bar{\boldsymbol{\mu}})$	$\bar{\delta}(\mathcal{D}_r \cup \tilde{\mathcal{D}}_r, \bar{\boldsymbol{\mu}})$	$\delta_{\min}(\mathcal{D}_r, \bar{\boldsymbol{\mu}})$	$\delta_{\min}(\mathcal{D}_r \cup \tilde{\mathcal{D}}_r, \bar{\boldsymbol{\mu}})$
$\bar{\boldsymbol{\mu}}^1$	0.602	0.601	0.431	0.323
$\bar{\boldsymbol{\mu}}^2$	0.643	0.658	0.321	0.318
$\bar{\boldsymbol{\mu}}^3$	0.698	0.701	0.544	0.358

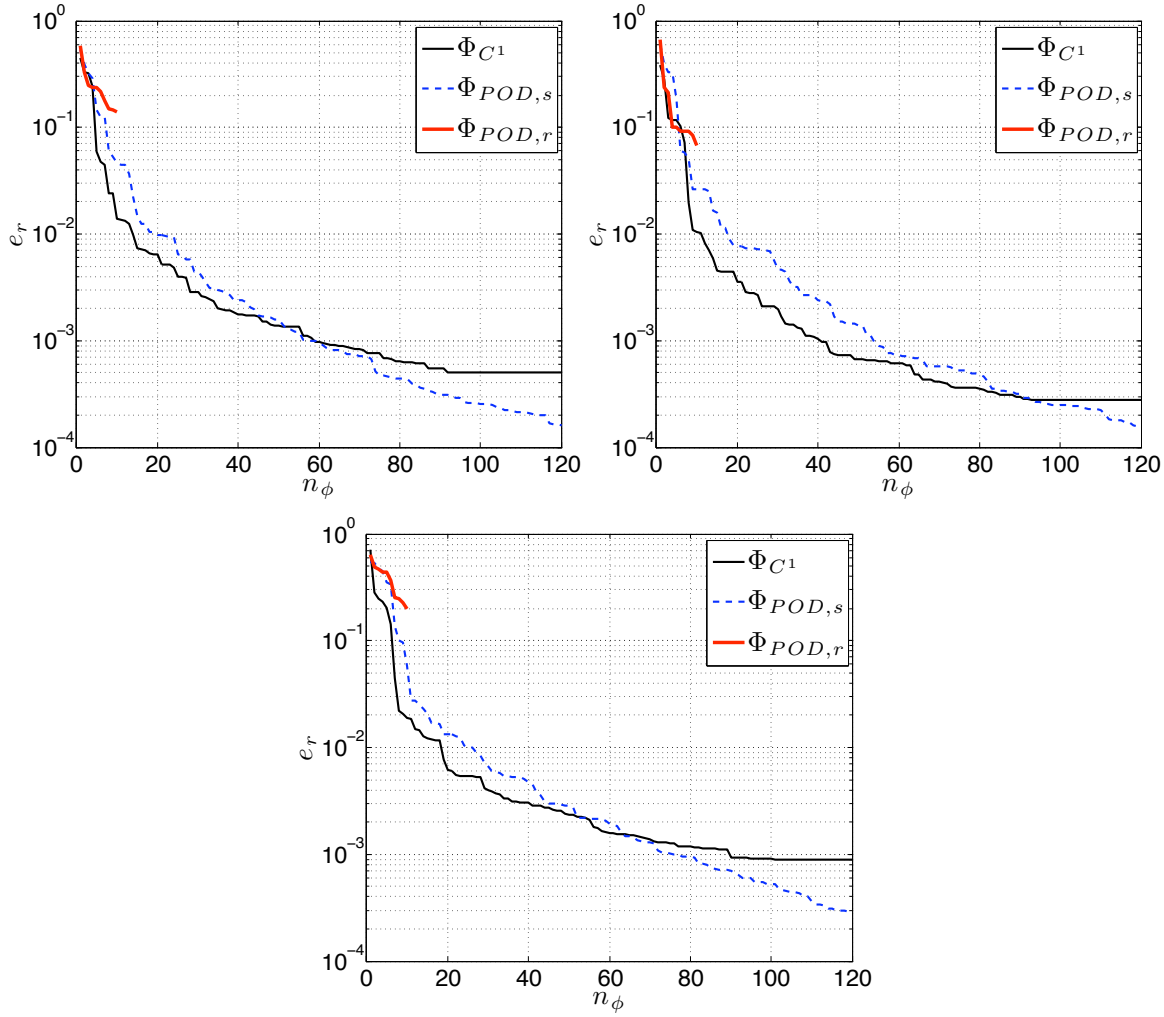


Figure 6. Dependence of relative error on n_ϕ, $n_r = 10$ for targets $\bar{\boldsymbol{\mu}}^1$, $\bar{\boldsymbol{\mu}}^2$, $\bar{\boldsymbol{\mu}}^3$ (clockwise from top left).

Table 8. Relative errors, $n_\phi = 10$, $n_r = 10$.

$\bar{\boldsymbol{\mu}}$	$e_r(\Phi_{C^1}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\text{POD},s}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\text{POD},r}, \bar{\boldsymbol{\mu}})$
$\bar{\boldsymbol{\mu}}^1$	1.38×10^{-2}	4.64×10^{-2}	14.06×10^{-2}
$\bar{\boldsymbol{\mu}}^2$	1.05×10^{-2}	2.63×10^{-2}	6.74×10^{-2}
$\bar{\boldsymbol{\mu}}^3$	1.88×10^{-2}	5.74×10^{-2}	19.87×10^{-2}

Table 9. Relative errors, $n_\phi = n_{\phi,\mathcal{E}}$, $n_r = 10$.

$\bar{\boldsymbol{\mu}}$	$e_r(\Phi_{C^1}, \bar{\boldsymbol{\mu}})$	$e_r(\Phi_{\text{POD},s}, \bar{\boldsymbol{\mu}})$	$n_{\phi,\mathcal{E}}$
$\bar{\boldsymbol{\mu}}^1$	0.73×10^{-2}	1.48×10^{-2}	15
$\bar{\boldsymbol{\mu}}^2$	1.03×10^{-2}	2.63×10^{-2}	11
$\bar{\boldsymbol{\mu}}^3$	1.26×10^{-2}	2.33×10^{-2}	14

V. Concluding Remarks

In this paper, we have introduced the Compact POD basis, which combines ideas from the Hermite subspace and the POD method of model reduction. This basis produces an optimization-oriented reduced-order model (ROM) that is inherently more robust, less expensive to construct, and utilizes many fewer reference configurations that the typical POD approach. We have also presented an extension of adaptive sampling procedures to the optimization of static systems, where CPOD is a critical component. Finally, we have applied the method to the analysis of a plane elasticity problem, where we observed excellent results in terms of accuracy and cost compared with the standard approach.

Future work includes implementing algorithm 2 for the optimization of static systems using CPOD, investigating the performance of CPOD of order 2 and higher, and efficiently solving the reduced system in Eq. (6), which is difficult in the presence of operators with non-affine parameter dependence.[g]

Finally, we note that the extension of CPOD to dynamical systems is straightforward, since the snapshots in this case correspond to the state vector and sensitivities at various configurations and at different times or frequencies. The same snapshot weighting strategy applies.

Acknowledgments

The authors thank Kurt Maute and his research group for providing SDESIGN, which has been a valuable tool for defining shape parameters in our research. Kevin Carlberg gratefully acknowledges the invaluable consultation of David Amsallem, Julien Cortial, and Jacob Mattingley, as well as the financial support of the Department of Defense NDSEG fellowship and the National Science Foundation Graduate Research Fellowship.

References

[1]LeGresley, P. and Alonso, J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition," *AIAA Paper 2000-2545, Fluids 2000 Conference and Exhibit, Denver, CO*, June 19–22, 2000.

[2]Oliveira, I. and Patera, A., "Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations," *Optimization and Engineering*, Vol. 8, No. 1, 2007, pp. 43–65.

[3]Afanasiev, K. and Hinze, M., "Adaptive control of a wake flow using proper orthogonal decomposition," *Shape Optimization and Optimal Design: Proceedings of the IFIP Conference*, Vol. 216, 2001, pp. 317–332.

[4]Ravindran, S., "A reduced-order approach for optimal control of fluids using proper orthogonal decomposition," *International Journal for Numerical Methods in Fluids*, Vol. 34, No. 5, 2000, pp. 425–448.

[5]Arian, E., Fahl, M., and Sachs, E. W., "Trust-region proper orthogonal decomposition for flow control," Tech. Rep. 25, ICASE, 2000.

[6]Kunisch, Karl and Volkwein, Stefan , "Proper orthogonal decomposition for optimality systems," *Mathematical Modelling and Numerical Analysis*, Vol. 42, No. 1, January 2008, pp. 1–23.

[7]Ganapathysubramanian, S. and Zabaras, N., "Design across length scales: a reduced-order model of polycrystal plasticity for the control of microstructure-sensitive material properties," *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, No. 45–47, 2004, pp. 5017–5034.

[8]Huynh, D. and Patera, A., "Reduced basis approximation and a posteriori error estimation for stress intensity factors," *International Journal for Numerical Methods in Engineering*, Vol. 72, No. 10, 2007, pp. 1219–1259.

[9]Burkardt, J., Gunzburger, M., and Webster, C., "Reduced Order Modeling of some Nonlinear Stochastic Partial Differential Equations," *International Journal of Numerical Analysis and Modeling*, Vol. 4, No. 3–4, 2007, pp. 368–391.

[10]Weickum, G., Eldred, M., and Maute, K., "Multi-point Extended Reduced Order Modeling For Design Optimization and Uncertainty Analysis," *AIAA Paper 2006-2145, 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2nd AIAA Multidisciplinary Design Optimization Specialist Conference), Newport, RI*, May 1–4, 2006.

[11]Lieu, T., Farhat, C., and Lesoinne, M., "Reduced-order fluid/structure modeling of a complete aircraft configuration," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 41–43, 2006, pp. 5730–5742.

[12]Amsallem, D., Farhat, C., and Lieu, T., "Aeroelastic Analysis of F-16 and F-18/A Configurations Using Adapted CFD-Based Reduced-Order Models," *AIAA Paper 2007-2364, 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, June 1–4, 2007.

[13]Amsallem, D. and Farhat, C., "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, Vol. 46, No. 7, July 2008, pp. 1803–1813.

[14]Tang, K., Graham, W., and Peraire, J., "Active flow control using a reduced order model and optimum control," *AIAA Paper 1996-1946, Fluid Dynamics Conference, 27th, New Orleans, LA*, June 17–20, 1996.

[g]Empirical interpolation[39] is one promising approach to avoid this problem.

[15]Banks, H., del Rosario, R., and Smith, R., "Reduced-order model feedback control design: numerical implementation in a thin shell model," *IEEE Transactions on Automatic Control*, Vol. 45, No. 7, 2000, pp. 1312–1324.

[16]Ito, K. and Ravindran, S., "A reduced-order method for simulation and control of fluid flows," *Journal of Computational Physics*, Vol. 143, No. 2, 1998, pp. 403–425.

[17]Atwell, J. and King, B., "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations," *Mathematical and Computer Modelling*, Vol. 33, No. 1–3, 2001, pp. 1–19.

[18]Bergmann, M., Cordier, L., and Brancher, J.-P., "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model," *Physics of Fluids*, Vol. 17, No. 9, 2005, pp. 097101.

[19]Banks, H., Joyner, M., Winchesky, B., and Winfree, W., "Nondestructive evaluation using a reduced-order computational methodology," *Inverse Problems*, Vol. 16, No. 4, 2000, pp. 929–945.

[20]Grepl, M., Nguyen, N., Veroy, K., Patera, A., and Liu, G., "Certified rapid solution of partial differential equations for real-time parameter estimation and optimization," *Proceedings of the 2nd Sandia Workshop of PDE-Constrained Optimization*, SIAM Computational Science and Engineering Book Series, 2005.

[21]Wang, J. and Zabaras, N., "Using Bayesian statistics in the estimation of heat source in radiation," *International Journal of Heat and Mass Transfer*, Vol. 48, No. 1, 2005, pp. 15–29.

[22]Sirovich, L., "Turbulence and the dynamics of coherent structures, I-III," *Quarterly of Applied Mathematics*, Vol. 45, No. 3, 1987, pp. 561–590.

[23]Kunisch, K. and Volkwein, S., "Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition," *Journal of Optimization Theory and Applications*, Vol. 102, No. 2, 1999, pp. 345–371.

[24]Rozza, G., Huynh, D., and Patera, A., "Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations," *Archives of Computational Methods in Engineering*, Vol. 15, No. 3, September 2008, pp. 229–275.

[25]McKay, M., Beckman, R., and Conover, W., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[26]Du, Q., Faber, V., and Gunzburger, M., "Centroidal voronoi tesselations," *Siam Review*, Vol. 41, No. 4, 1999, pp. 637–676.

[27]Veroy, K., Prud'homme, C., Rovas, D., and Patera, A., "A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations," *AIAA Paper 2003-3847, 16th AIAA Computational Fluid Dynamics Conference*, June 23–26, 2003.

[28]Bui-Thanh, T., Willcox, K., and Ghattas, O., "Model reduction for large-scale systems with high-dimensional parametric input space," *AIAA Paper 2007-2049, 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference*, June 1–4, 2007.

[29]Golub, G. and Van Loan, C., *Matrix Computations*, Johns Hopkins University Press, 1996.

[30]Rathinam, M. and Petzold, L. R., "A New Look at Proper Orthogonal Decomposition," *SIAM Journal on Numerical Analysis*, Vol. 41, No. 5, 2003, pp. 1893–1925.

[31]Noor, A. and Peters, J., "Reduced basis technique for nonlinear analysis of structures," *AIAA Journal*, Vol. 18, No. 4, 1980, pp. 455–462.

[32]Peterson, J., "The Reduced Basis Method for Incompressible Viscous Flow Calculations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, 1989, pp. 777.

[33]Farhat, C., Crivelli, L., and Roux, F., "Extending substructure based iterative solvers to multiple load and repeated analyses," *Computer Methods in Applied Mechanics and Engineering*, Vol. 117, No. 1–2, July 1994, pp. 195–209.

[34]Haftka, R. and Adelman, H., "Recent developments in structural sensitivity analysis," *Structural and Multidisciplinary Optimization*, Vol. 1, No. 3, September 1989, pp. 137–151.

[35]Nocedal, J. and Wright, S., *Numerical Optimization*, Springer, 1999.

[36]Szabo, B. and Babuška, I., *Finite Element Analysis*, Wiley-Interscience, 1991.

[37]Maute, K. and Raulli, M., *FEM—Optimization Module and SDESIGN User Guides*, 0.2 ed., June 2006.

[38]Homberg, D. and Volkwein, S., "Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition," *Mathematical and Computer Modelling*, Vol. 38, No. 10, November 2003, pp. 1003–1028.

[39]Barrault, M., Maday, Y., Nguyen, N., and Patera, A., "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations," *Comptes rendus-Mathématique*, Vol. 339, No. 9, 2004, pp. 667–672.